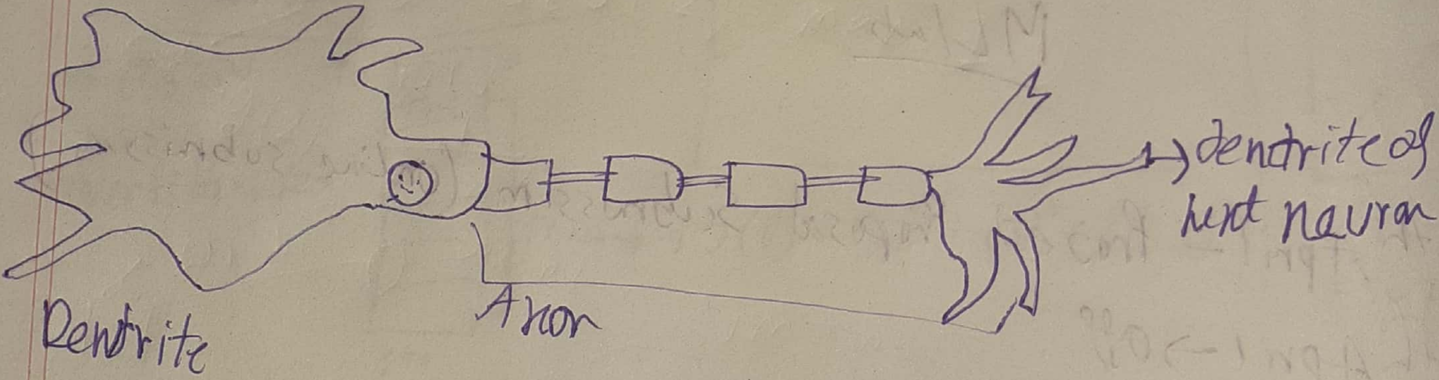
15/3/23

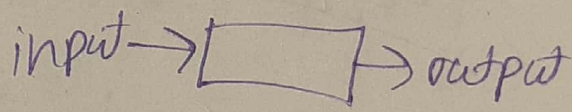
ML (T-11)

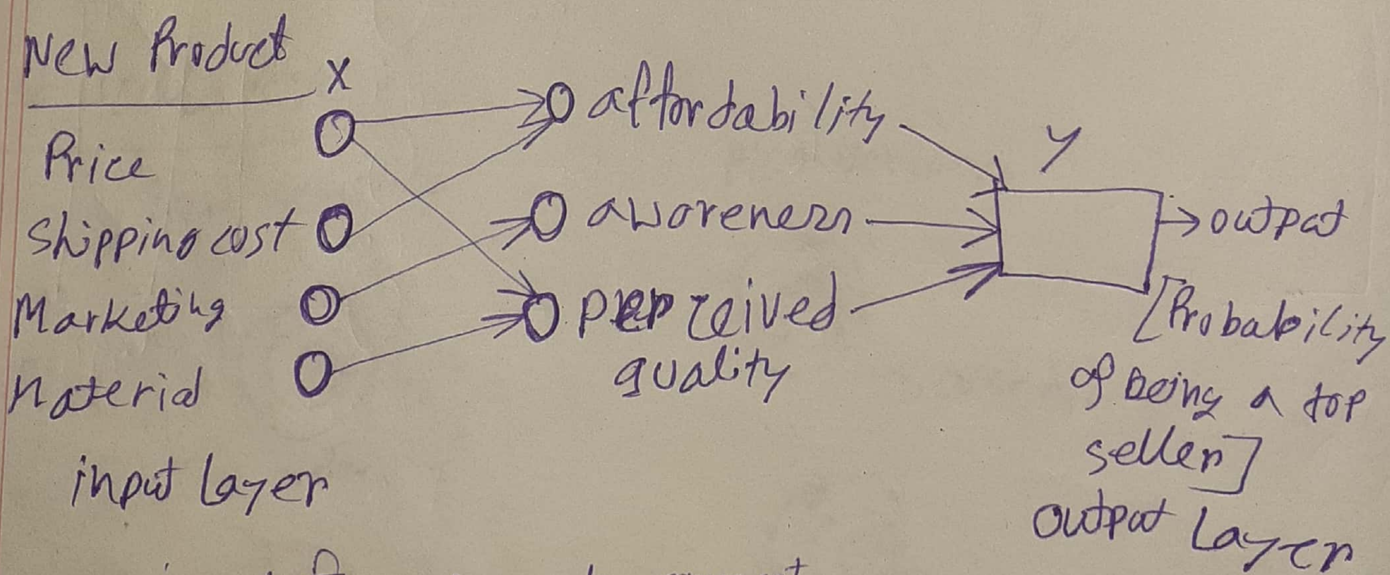
Neural Network

\# **Neural Network:-** When a machine try to work like human brain using algorithm then it is called Neural Network.

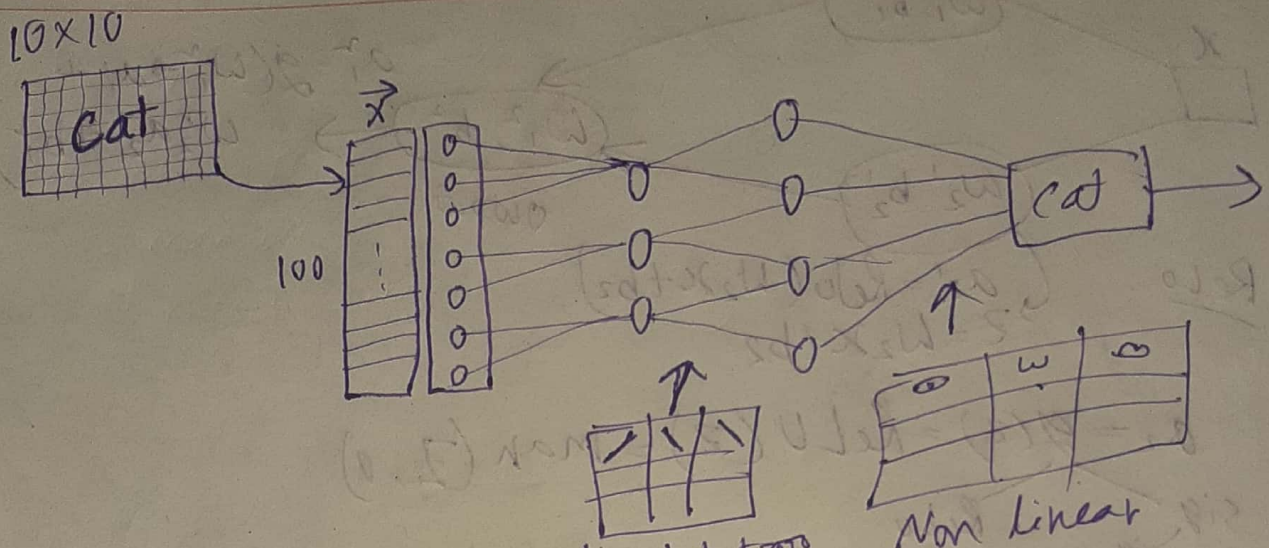


Neuron

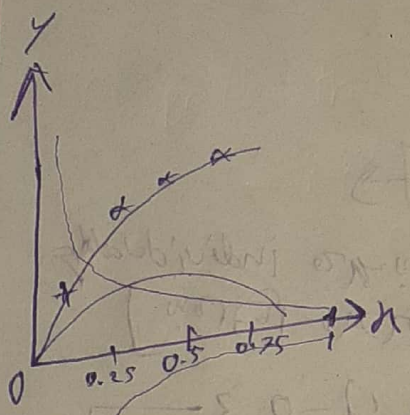


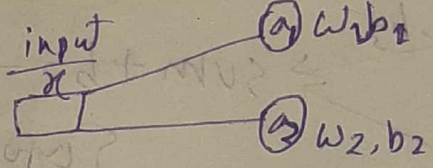


→ শুধু input দিলে ও auto connect করে।



$a_1 = wx + b$ (w: weight, b: bias)



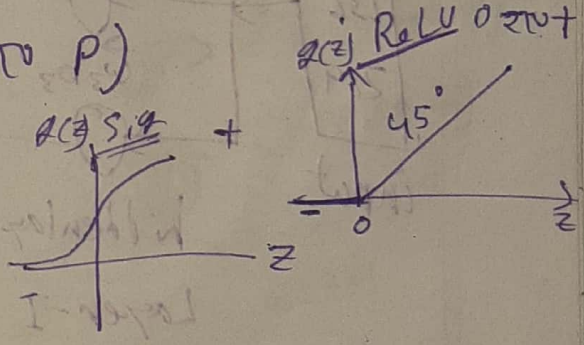


Linear Activation function → Linear Regression (P, N)

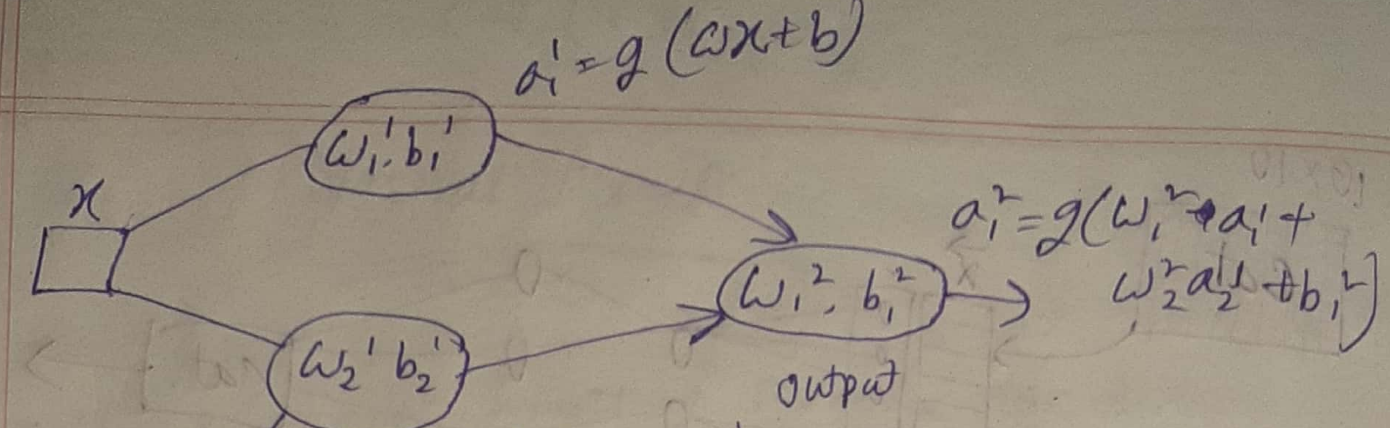
Sigmoid:- $g(z) = \frac{1}{1+e^{-z}}$ → binary classification (P)

ReLU:- $g(z) = \max(z, 0)$ → (0 TO P)

tanh:- $\frac{e^z - e^{-z}}{e^z + e^{-z}}$ (self study)



→ Non linear use 2



ReLU

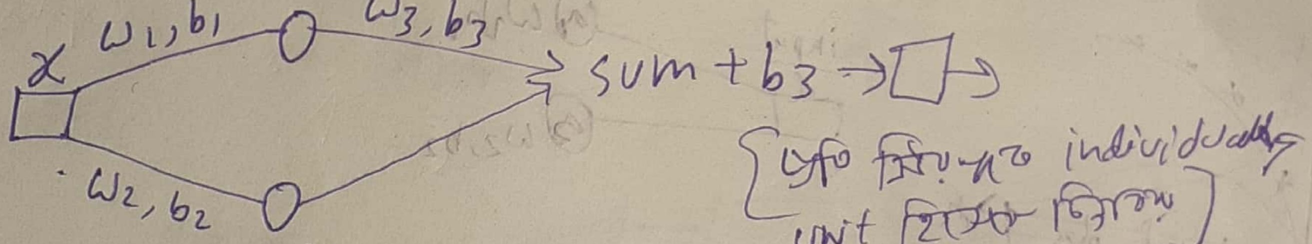
$a_2^1 = ReLU(W_2 x + b_2)$

$z = W_2 x + b_2$

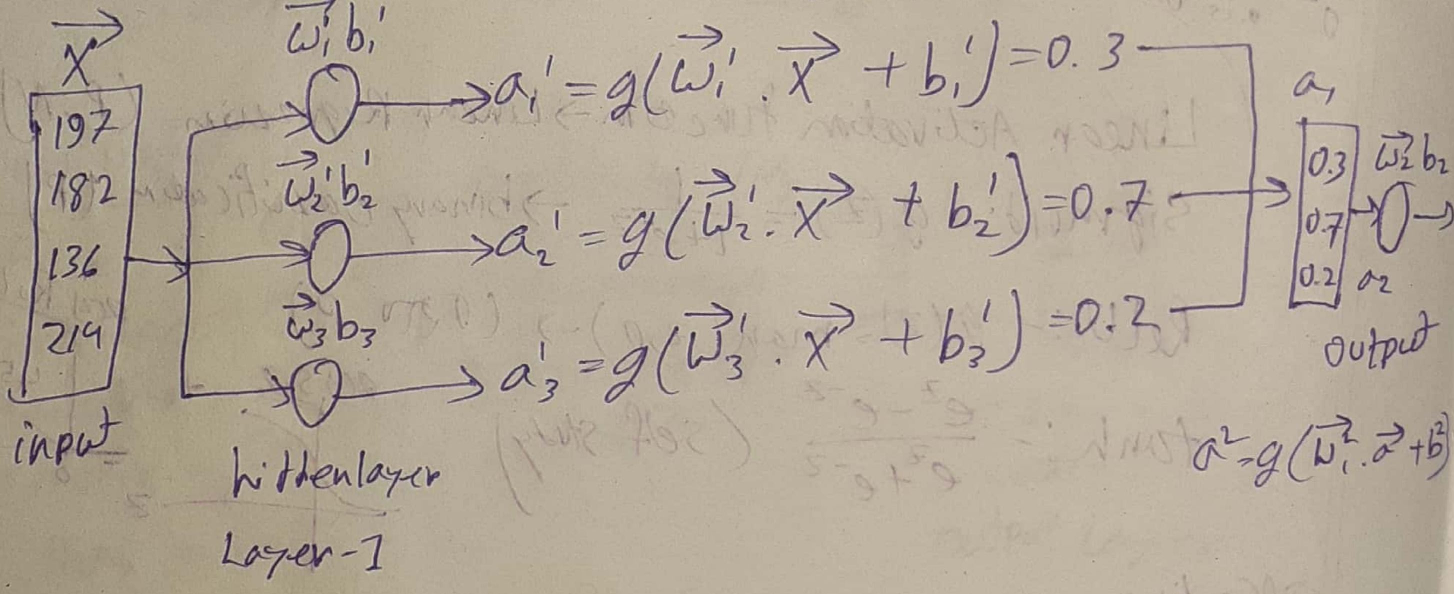
$a_2 = g(z) = ReLU(z) = max(z, 0)$

sig — ReLU

$g(z) = \frac{1}{1+e^{-z}}$



{প্রতি নিউরনকে individually unit হিসেবে নিয়েছি}



a? value হয় হল or hidden layer এর উপর depend করে,

ML (T-12)

12/4/23

→ Binary = sygmoid function.

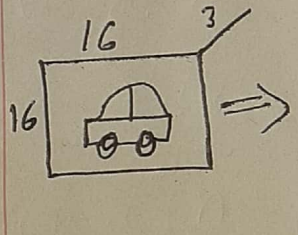
Neuron Network

Neuron = Linear + activation

Model = architecture + parameters.

Logistics Regression:- sigmoid function (Activation)

Goal 1:- Find cars in image → [1 = Present, 0 = Absent]



$x_1, x_2, \ldots, x_n$ → $wx+b$ → $\hat{y} = \sigma(\theta^T x)$

$= \sigma(wx+b)$

(1,1) (1,768) (16×16+3, 1) (768,1)

$n = 16 \times 16 + 3$

$= 768$

Nb of parameters for two neurons $= \underset{w}{768} + \underset{b}{1} = 769$

→ Every neuron has a bias (b)

→ Every edge has a weight (w)

Tranning Process:

① Initialize weights (w) & bias (b)

② Find the optimal w & b. → Loss function

$$\mathcal{L} = -[\underset{\text{T.V.}}{y} \log \underset{\text{P.V.}}{\hat{y}} + (1-y)\log(1-\hat{y})]$$

③ use $\hat{y} = \sigma(wX+b)$ to predict.

For minimization:-

$$w = w - \alpha \frac{\partial \mathcal{L}}{\partial w}$$

$$b = b - \alpha \frac{\partial \mathcal{L}}{\partial b}$$

} Gradient Decent

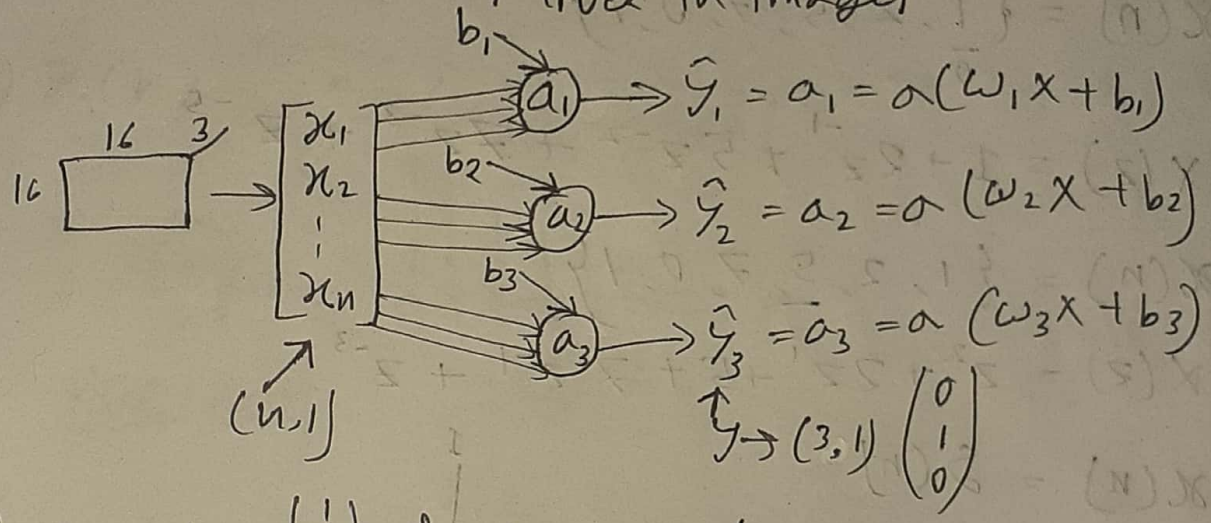
16/4/23

Neural Network

Logistic Regression

goal 1:- Find cars in images → 1G 16 → $\hat{y} = \sigma(wx+b)$

goal 2:- Find car/bus/truck in images



for cars, $y_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}$, for bus, $y_2 = \begin{pmatrix}0\\1\\0\end{pmatrix}$, for truck, $y_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}$

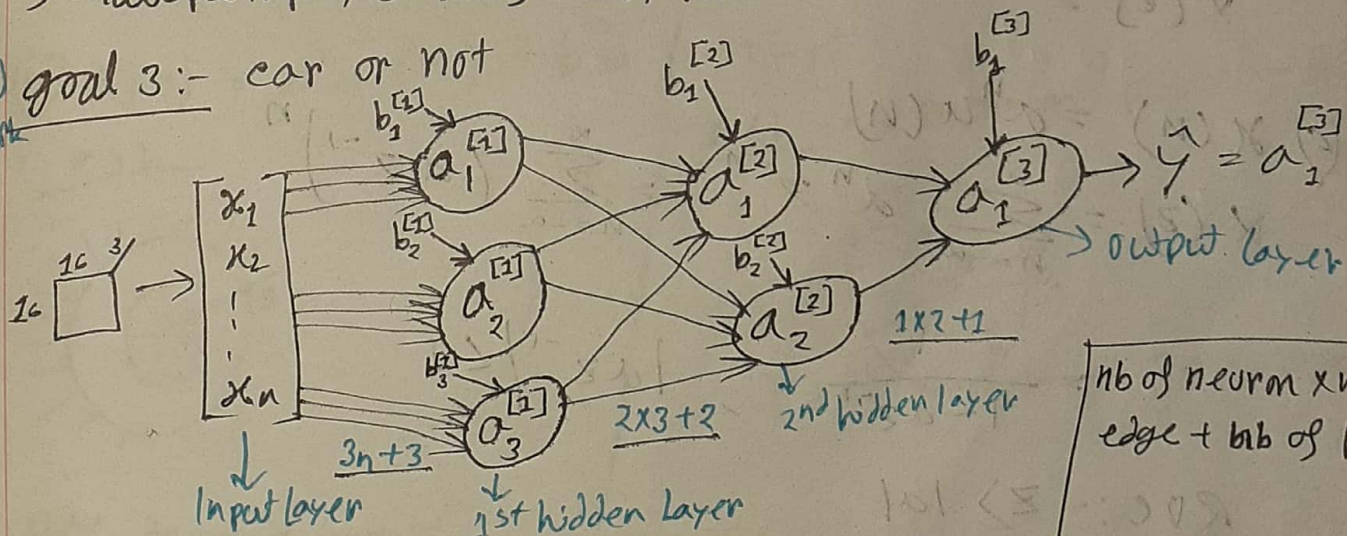
Parameter = $\underset{w}{3n} + \underset{b}{3}$

loss function:- $\mathcal{L} = -\sum_{k=1}^{3}\left[y_k \log \hat{y}_k + (1-y_k)\log(1-\hat{y}_k)\right]$

→ single input/element = loss function.
→ Multiple input/elements = cost function.

Neural Network

goal 3:- car or not



nb of neuron x nb of edge + nb of bias.

total Parameter = 3n+3+8+3

ML (T-14)

4/5/23

Online

NW

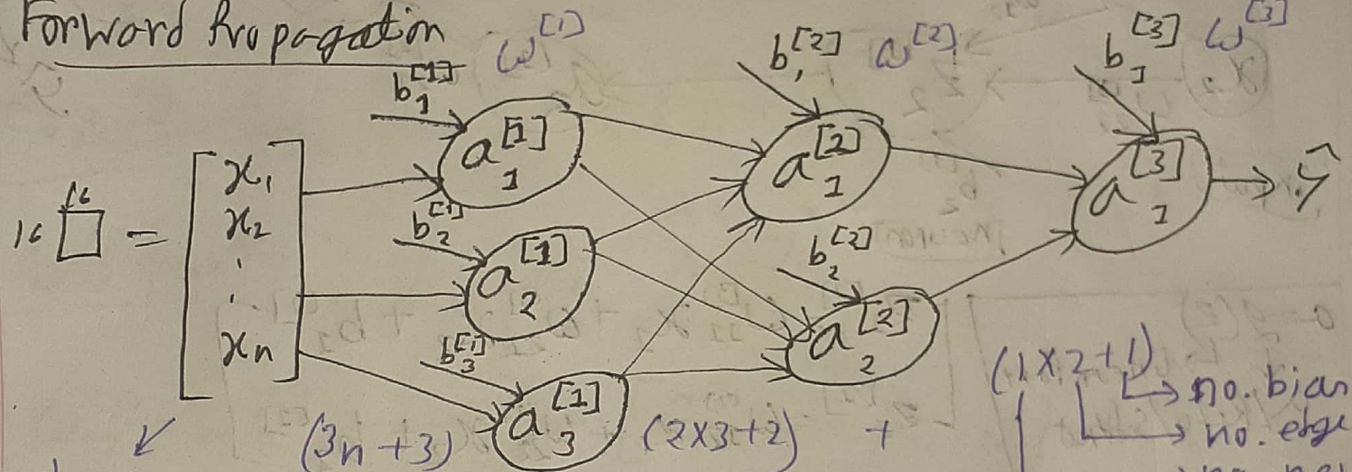
neuron = Linear + Activation

Linear → straight line $wx + b$

Activation → sigmoid, ReLU, etc

model = architecture + Parameters.

Forward Propagation



(1×2+1) → no. bias; → no. edge; → no. neuron

↓ training phase

→ এভাবে প্রতি node এর value বের করি এভাবে FP use হয়।

$$a_1^{[1]} = \sigma(z) = \sigma(wx + b)$$

Forward Propagation :- Left to Right = FP.

size

$(3,1) \rightarrow z^{[1]} = w^{[1]} x + b^{[1]}$ — sizes: $w^{[1]}$ (3,n), x (n,1), $b^{[1]}$ (3,1), $w^{[1]}x$ (3,1)

$(3,1) \rightarrow a^{[1]} = \sigma(z^{[1]}) = g(z^{[1]})$ — g: activation function

$(2,1) \rightarrow z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$ — sizes: $w^{[2]}$ (2,3), $a^{[1]}$ (3,1), $b^{[2]}$ (2,1), $w^{[2]}a^{[1]}$ (2,1)

$(2,1) \rightarrow a^{[2]} = \sigma(z^{[2]})$

$(1,1) \rightarrow z^{[3]} = w^{[3]} a^{[2]} + b^{[3]}$ — sizes: $w^{[3]}$ (1,2), $a^{[2]}$ (2,1), $b^{[3]}$ (1,1), $w^{[3]}a^{[2]}$ (1,1)

$(1,1) \rightarrow a^{[3]} = \sigma(z^{[3]}) \rightarrow \hat{y}$

এখানে w কেন আগের আসবে না,

$w^{[2]} a^{[1]} + b^{[2]}$

$w^{[2]}$ → self layer এর w

$a^{[1]}$ → privious layer এর value

$b^{[2]}$ → self bias

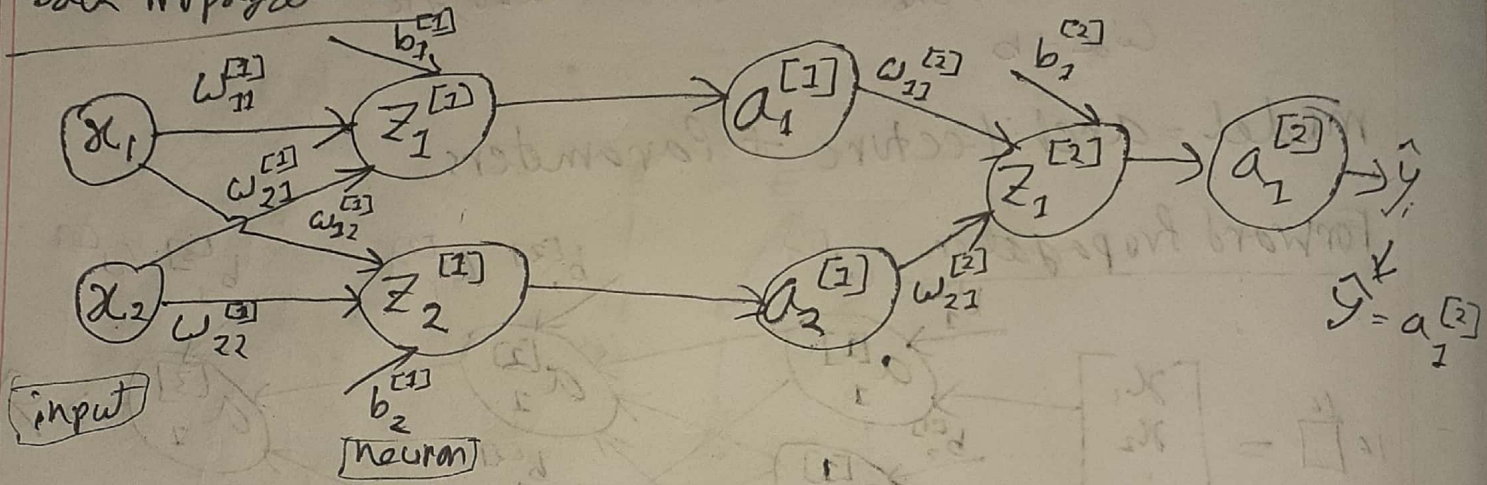
ML (T-15)

21/5/23

NN

Forward Propagation:- generate output. (node update)

Back Propagation:- Minimize cost function. (w update)

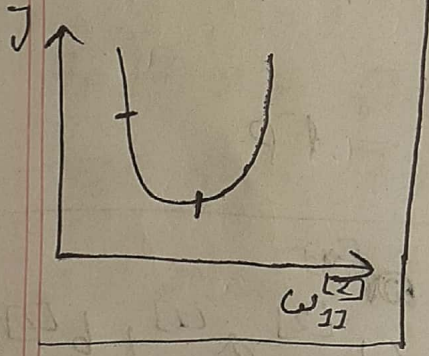


$a = g(z)$ → Linear Part
↓
σ/Relu/Softmax

$$z_1^{[1]} = w_{11}^{[1]} x_1 + w_{21}^{[1]} x_2 + b_1^{[1]}$$

$$z_2^{[1]} = w_{12}^{[1]} x_1 + w_{22}^{[1]} x_2 + b_2^{[1]}$$

$$w_{11}^{[2]} = w_{11}^{[2]} - \alpha \frac{\partial J}{\partial w_{11}^{[2]}}$$



$$a_1^{[1]} = \sigma(z_1^{[1]})$$

$$a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_1^{[2]} = w_{11}^{[2]} a_1^{[1]} + w_{21}^{[2]} a_2^{[1]} + b_1^{[2]}$$

$$a_1^{[2]} = \hat{y}_i = \sigma(z_1^{[2]})$$

Back Propagation:- aims to minimize the cost function by adjusting network parameters.

cost = multiple training data.
lost = single training data.

if $y = \sigma(z) = \frac{1}{1+e^{-z}}$

then $\frac{\partial y}{\partial z} = (1+e^{-z})^{-2} e^{-z} = \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}}$ ($e^{-z} \to 1 - \frac{1}{1+e^{-z}} = \frac{1+e^{-z}-1}{1+e^{-z}}$)

$= y \cdot (1-y)$

$a = \sigma(z)$

$\frac{\partial a}{\partial z} = a(1-a)$

Cost function :- $J = \sum_{i=1}^{N} \frac{1}{2} (\hat{y}_i - y_i)^2$

Loss function :- $\mathcal{L}_i = \frac{1}{2} (\hat{y}_i - y_i)^2$

$w_{11}^{[2]} = w_{11}^{[2]} - \alpha \frac{\partial \mathcal{L}_i}{\partial w_{11}^{[2]}}$ [Gradient descent]

$\frac{\partial \mathcal{L}_i}{\partial w_{11}^{[2]}}$:- $a_1^{[1]}$ —$w_{11}^{[2]}$→ $a_1^{[2]}$ / $\hat{y}_i$

from $a_1^{[1]}$ to $a_1^{[2]}$ / $\hat{y}_i$

$= \frac{\partial \mathcal{L}_i}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_1^{[2]}} \cdot \frac{\partial z_1^{[2]}}{\partial w_{11}^{[2]}}$

$\frac{\partial \mathcal{L}_i}{\partial \hat{y}_i} = (\hat{y}_i - y_i)$, $\frac{\partial \hat{y}_i}{\partial z_1^{[2]}} = \hat{y}_i(1-\hat{y}_i)$, $\frac{\partial z_1^{[2]}}{\partial w_{11}^{[2]}} = a_1^{[1]}$

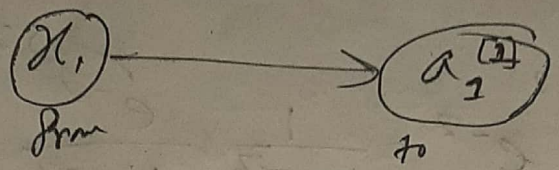
$(\hat{y}_i - y_i)\,\hat{y}_i(1-\hat{y}_i) = \delta \hat{y}_i$

$= a_1^{[1]} \delta \hat{y}_i$

chain rule

output → z → w

$$\frac{\partial z_i}{\partial w_{21}^{[1]}} :$$



$$= \underbrace{\frac{\partial z_i}{\partial \hat{y}_i}}_{(\hat{y}_i - y_i)} \cdot \underbrace{\frac{\partial \hat{y}_i}{\partial z_1^{[2]}}}_{\hat{y}_i(1-\hat{y}_i)} \cdot \underbrace{\frac{\partial z_1^{[2]}}{\partial a_1^{[1]}}}_{w_{11}^{[2]}} \cdot \underbrace{\frac{\partial a_1^{[1]}}{\partial z_1^{[1]}}}_{a_1^{[1]}(1-a_1^{[1]})} \cdot \underbrace{\frac{\partial z_1^{[1]}}{\partial w_{21}^{[1]}}}_{x_2}$$

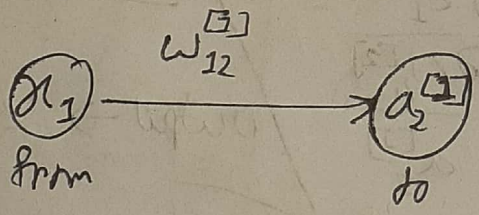
$$\underbrace{\qquad\qquad\qquad}_{\partial \hat{y}_i}$$

$$= x_2 \cdot a_1^{[1]}(1-a_1^{[1]})\, w_{11}^{[2]} \cdot \underbrace{\partial \hat{y}_i}_{\partial a_1^{[1]}}$$

$$= x_2 \cdot a_1^{[1]}(1-a_1^{[1]})\, \partial a_1^{[1]}$$

↙ from ↓ to $(1-\text{to})$ ↙ δ_{to}

$$\frac{\partial z_i}{\partial w_{22}^{[1]}} = x_1 a_2^{[1]}(1-a_2^{[1]}) \underbrace{\partial a_2^{[1]}}_{w_{21}^{[2]}\, \partial \hat{y}_i}$$

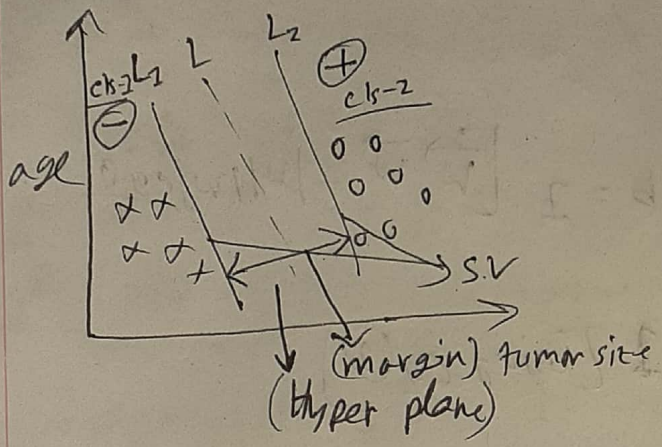


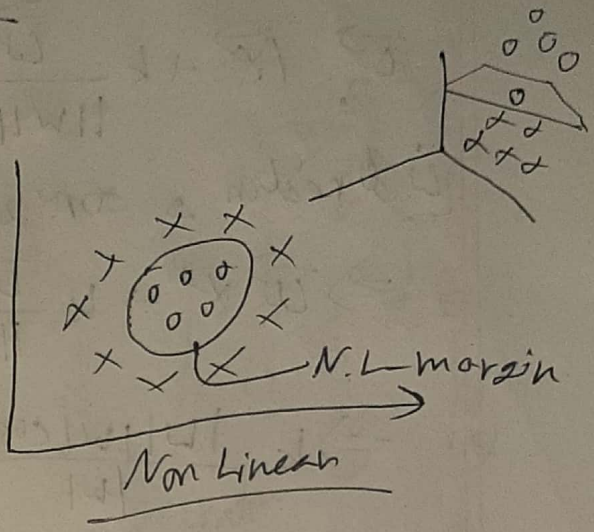
ML (T-16)

10/5/23

Support Vector Machine (SVM):-

→ Classifer





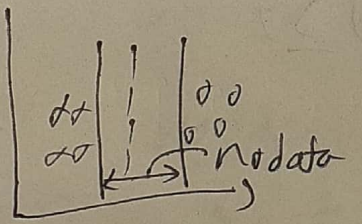
Linear

→ SVM is a margin classifer

→ Type → Linear SVM
→ Non Linear SVM

Another type → Hard Margin SVM
→ Soft Margin SVM

① Hard Margin SVM:-



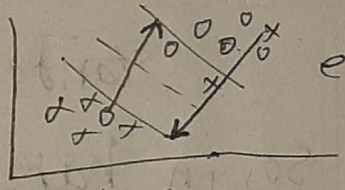
$$\begin{bmatrix} \vec{x_1} \\ \vec{x_2} \\ \vdots \\ \vec{x_n} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad y = [+1, -1]$$

data = n
features = m

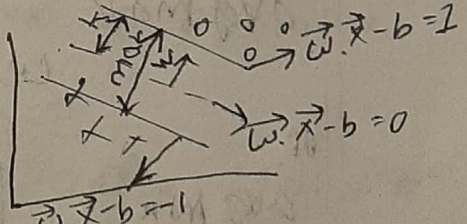
$$\vec{\omega} \cdot \vec{x} - b = 0 \quad \text{---(1)}$$

$$\Rightarrow \omega_1 \cdot x_1 + \omega_2 \cdot x_2 + \dots + \omega_m \cdot x_m - b = 0$$

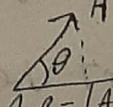
② Soft Margin SVM:-



$\omega, b = ?$
→ margin max



$A \cdot B = A \times B_1 + A \times B_2 + \dots$



$A \cdot B = |A||B| \cos\theta$
$= |A| \cos\theta \, |B|$

After moving k units towards w direction we get

$$\vec{w}\cdot\left(\vec{x} + k\frac{\vec{w}}{\|w\|}\right) - b = 1$$

($\vec{w}$ direction → unit vector)

[$\vec{w}$ direction এ যাচ্ছে]

$$\Rightarrow \vec{w}\cdot\vec{x} + k\frac{\vec{w}\cdot\vec{w}}{\|w\|} - b = 1 \quad \left[\vec{w}\cdot\vec{w} = |w||w|\cos 0 = 1\right]$$

$$\Rightarrow k\frac{|w||w|\cos\theta^\circ}{\|w\|} + 0 = 1 \quad [\text{using eq } (1)]$$

$$\Rightarrow k = \frac{1}{|w|}$$

Margin size $= 2k = \frac{2}{\|w\|}$ [যদি $w = \min$, তাহলে $k = \max$]

Max margin $\Rightarrow \min |w|$

constraints:- for all $y_i = 1$. $\vec{w}\cdot\vec{x_i} - b \geq 1$

for all $y_i = -1$. $\vec{w}\cdot\vec{x_i} - b \leq -1$

standard form $y_i = (\vec{w}\cdot\vec{x_i} - b) \geq 1$

So, in hard margin SVM:-

$$\arg\max_{w,b} \frac{2}{\|w\|}, \text{ such that } y_i(\vec{w}\vec{x_i} - b) \geq 1$$

or

$$\arg\min_{w,b} |w|, \text{ such that}$$

Support vectors:- $y_i(\vec{w}\vec{x_i} - b) = 1$

ML (T-17) 14/5/23

SVM Margin - max → w & b ও dpn করে.

$\underset{w,b}{\text{argmin}} \|w\|$ such that $y_i (w \cdot x_i - b) \geq 1, \forall_i = 1, \ldots, N$

y_i → true

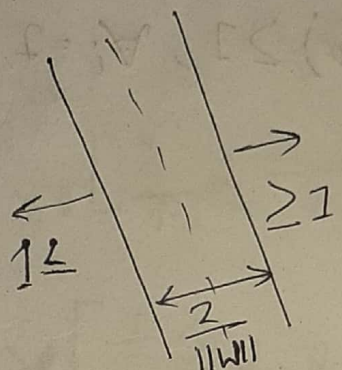
$(w \cdot x_i - b)$ → Prediction

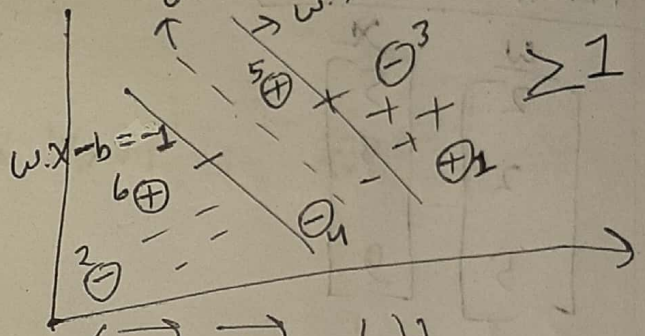
[margin এ কোন error থাকবে না]

[Hard margin → error নেই]

$\underset{w,b}{\text{MIN}} \|w\|$

$\text{Max} \frac{2}{\|w\|}$ ← margin





Soft Margin SVM:-

Hinge loss = $\max(0, 1 - y_i (\vec{w} \cdot \vec{x_i} - b))$

1st Point:- $HL = \max(0, 1 - 1(>1)) = 0$

2nd Point:- $HL = \max(0, 1 + 1(<-1)) = 0$

3rd Point:- $HL = \max(0, 1 + 1(>1)) = >1$

5th Point:- $HL = \max(0, 1 - 1(\text{between } 0 \text{ \& } 1)) = \text{between } 0 \text{ \& } 1$

6th Point:- $HL = \max(0, 1 - 1(<-1)) = >1$

Soft margin SVM:-

$\underset{w,b}{\text{argmin}} \frac{1}{N} \sum_{i=0}^{N} \max(0, 1 - y_i (wx_i - b)) + \lambda \|w\|^2$

↑ hyper parameter

SVM DUAL

Original Form (Primal)

$MIN_{w,b} \frac{1}{2}\|w\|^2$, such that

$y_i(\underbrace{w_i x}_{W^T \cdot x_i} - b) \geq 1, \forall_i = 2, \ldots N$

$\|W\| \cong \frac{1}{2}\|W\|^2$

w = $\begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}$ x = $\begin{bmatrix} 5 \\ 3 \\ 9 \end{bmatrix}$

$W^T x$

$\|W\|^2 = W.W = W^T.W$

Lagrangian:

$$\mathcal{L}(w,b,x) = \underbrace{\frac{1}{2} W^T W}_{\frac{1}{2}\|w\|^2} - \sum_{i=1}^{N} \alpha_i (y_i (W^T x_i - b) - 1) \quad \text{---(1)}$$

where, α_i = lagrange Multiplier

SVM DUAL:- $MAX_{\alpha_i \geq 0} [MIN_{w,b} \mathcal{L}(w,b,\alpha)]$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow W - \sum_{i=1}^{N} \alpha_i y_i x_i = 0$$

$$\Rightarrow W = \sum_{i=1}^{N} \alpha_i y_i x_i \quad \text{---(2)}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\Rightarrow \alpha^T y = 0 \quad \text{---(3)}$$

Substituting (2) & (3) into (1)

$$\Rightarrow \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j = \underset{w,b}{MIN}\ \mathcal{L}(w, b, \alpha)$$

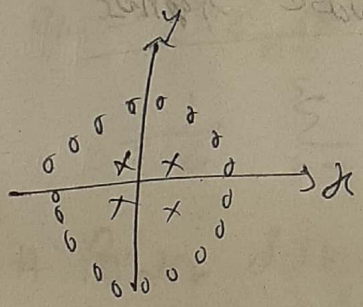
From SVM DUAL:-

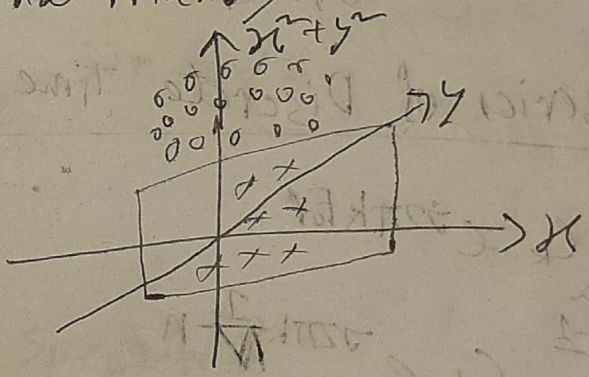
$$\underset{\alpha_i \geq 0}{MAX} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \right]$$

$$\underset{\alpha_i > 0}{MIN} \left[\frac{1}{2} \sum_{\substack{\text{Pairs} \\ SV(i,j)}} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{SV(i)} \alpha_i \right]$$

For datapoints those are not support vectors, $\alpha_i = 0$

→ Why DUAL:- Kernal Friendly





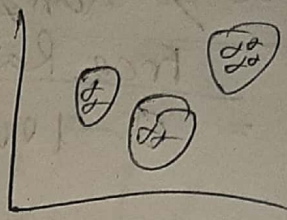
higher dimention

original (X) → T (X)

12/5/23

ML (T-18)

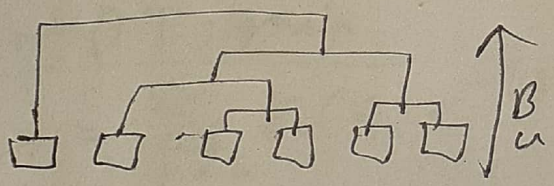
Unsupervised Learning

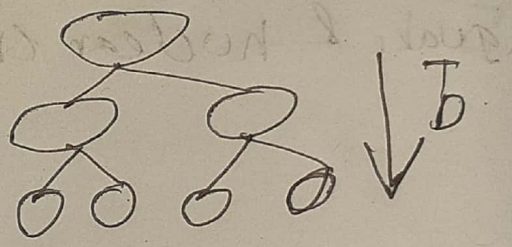


Clustering :

Clustering

- Hierarchical
  - Agglomerative (Bottom up - individual data)
  - Divisive (Top Down)
- Partitional
  - k means (no overlapping)
  - Fuzzy-C means (Overlapped cls)



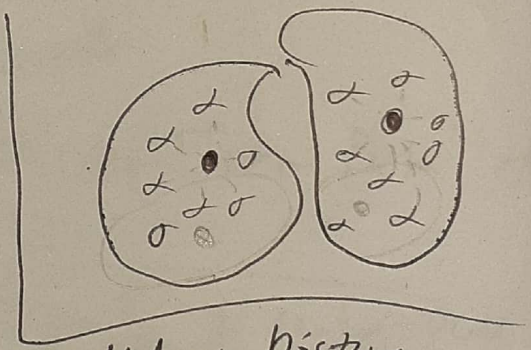


K-means clustering:

1. Select value of k = 2
2. Select k random points (centroids)
3. Assign each data point to nearest centroide.

$$\sqrt{(x_n - c_j)^2}$$

(x_n : data, c_j : centroids)



Euclidean Distance (each point to centroid)

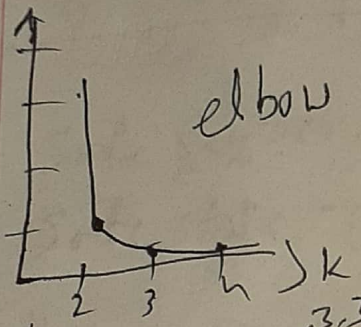
4. Find new cluster center by taking average.
5. Repeate steps 3 & 4 until none of the cluster center change.

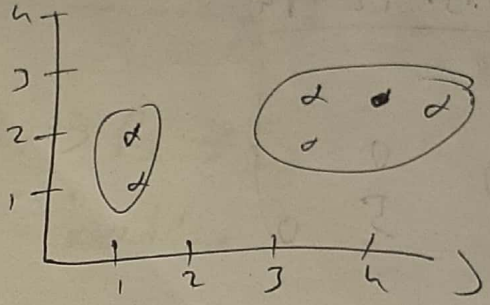
Selection of K: (Elbow method)

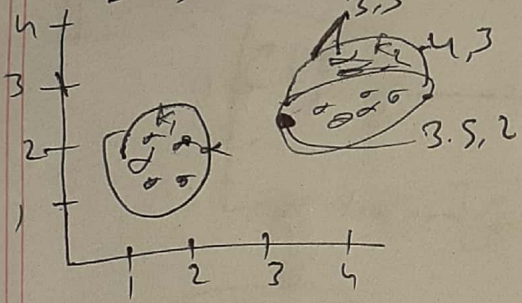
$k = 2, 3, 4, \ldots, 10$

Variance : sum of squares $SS = \sum_{n=1}^{n} (x_n - C_j)^2$

where, x_n = data points, C_j = center of the cluster in which data points belong





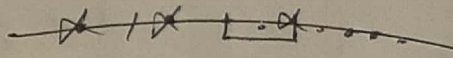


$k_1 = \frac{1+2}{2}, \frac{1+1}{2}$

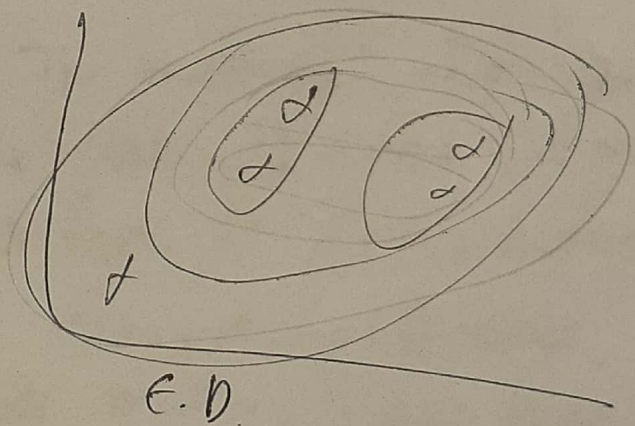
$= (1.5, 1)$

$k_2 = \frac{3+4}{2}, \frac{3+3}{2}$

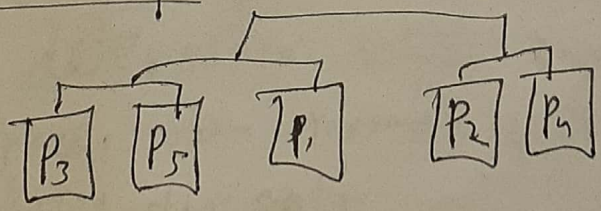
$= 3.5, 3$



	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2	8	0



cluster:



total - 4 clusters

	P_1	P_2	P_3, P_5	P_4
P_1	0			
P_2	9	0		
P_3, P_5	m(3,11) 3	m(7,10) 7	0	
P_4	6	5	m(9,8) 8	0

	P_1, P_3, P_5	P_2	P_4
P_1, P_3, P_5	0		
P_2	m(7,9) 7 / m(6,8) 6	0	
P_4		5	0

ML (T-19)   21/5/23

Dimensionality Reduction:-

→ convert higher dimentional data to lower dimention. (কম সংখ্যক dimention এর সাথে feature)

Motivation:-

→ Data Compression (training feature)

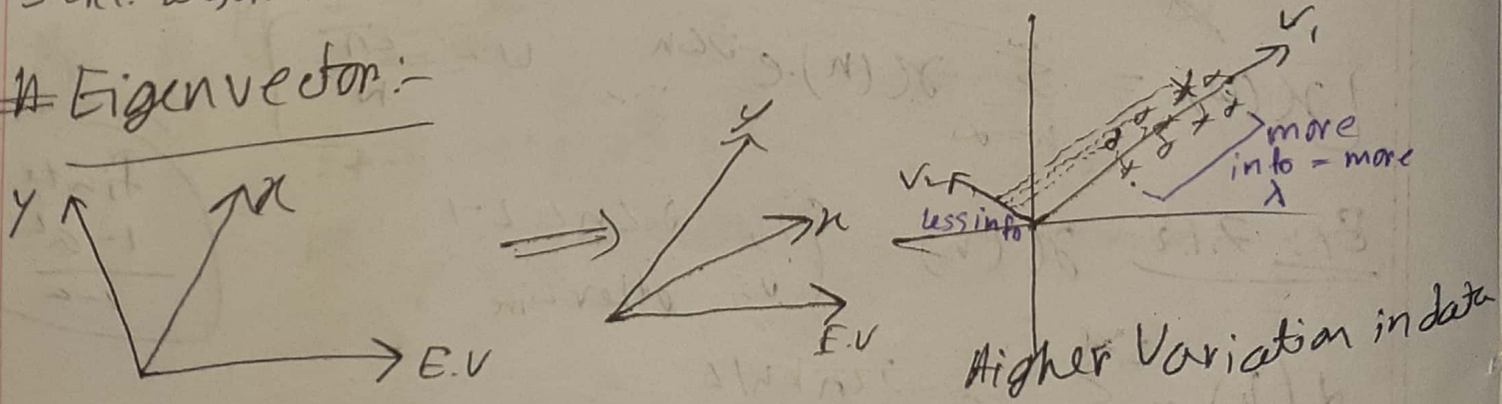
→ Visualization (1D, 2D)

PCA (Principal Component Analysis):- কোনগুলোর feature এ বেশি fit করে priority দিয়ে দেয় or Analysis.

CO-varience:- যখন input data ও তার different dimentation এর মধ্যে সম্পর্ক relation.

2D → 1D

Eigenvector:-



বড় λ = data এর info save করে.

λ ← Eigenvalue → more information

→ Eigen vector:- Eigenvector of a square matrix (A) is defined as a non-zero vector (V) by which, when a given matrix (A) is multiplied. it is equal to a scalar multiplier (λ) of that vector.

$$A.V = \lambda.V$$

where, A = given matrix, V = Eigenvector, λ = Eigen value.

→ E.V এ direction change হয় না, শুধু scalare হয় তার dimention এ থাকে porpindicular.

Ex:- $A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}$ $V = ?$, $\lambda = ?$

S/n:- $AV = \lambda V$

$\Rightarrow AV - \lambda V = 0$

$\Rightarrow AV - I\lambda V = 0$ [identity matrix, $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$]

$\Rightarrow V(A - I\lambda) = 0$

As $V \neq 0$, $A - I\lambda = 0$

Taking determinant

det $(A - I\lambda)$

$$\det\left(\begin{bmatrix}3 & 2\\ 2 & 6\end{bmatrix}-\begin{bmatrix}\lambda & 0\\ 0 & \lambda\end{bmatrix}\right)=0$$

$$\Rightarrow \det\left(\begin{bmatrix}3-\lambda & 2\\ 2 & 6-\lambda\end{bmatrix}\right)=0$$

$$\Rightarrow (3-\lambda)(6-\lambda)-4=0$$

$$\Rightarrow \lambda^2-3\lambda-6\lambda+18-4=0$$

$$\Rightarrow \lambda^2-9\lambda+14=0$$

$$\Rightarrow (\lambda-7)(\lambda-2)=0$$

$\lambda = 7, 2$

↳ Primary E.V

→ E. Value
→ E. Vector
→ dim

For $\lambda = 7$,

$$\begin{bmatrix}3 & 2\\ 2 & 6\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=7\begin{bmatrix}x\\ y\end{bmatrix}$$

$$\Rightarrow 3x+2y=7x \quad \text{(i)}$$

again $2x+6y=7y$ (ii)

from (i) − (ii)

$x=1,\ y=2$

$v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\lambda = 3$

Unit vector: Any vector can become a unit vector by dividing it by the magnitude of the vector.

$$\|v\| = \sqrt{1^2 + 2^2} = \sqrt{5}$$

Unit vector of $v = \frac{v}{\|v\|} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix}$

For $\lambda = 2$,

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 2 \begin{bmatrix} x \\ y \end{bmatrix}$$

$3x + 2y = 2x$ ——— (iii)

$2x + 6y = 2y$ ——— (iv)

by solving (iii) & (iv) we get: $x = 2, y = 1$

$v = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, unit vector $= \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}$

Data compression

PCA Given data, $M = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}$ convert it to 1D.

Sol: $M^T = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix}$

Co-variance Matrix, $M^T M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}$

$= \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix}$

$\begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} V - I\lambda V = 0$

$\Rightarrow \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} - I\lambda = 0 \quad [\lambda - 1.\lambda = 0]$

$\det \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = 0$

$\Rightarrow (30-\lambda)^2 = 28^2$

$\Rightarrow 30 - \lambda = \pm 28$

$\Rightarrow \lambda = 58 \& 2$

For $\lambda = 58$,

$$\begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 58 \begin{bmatrix} x \\ y \end{bmatrix}$$

$\Rightarrow 30x + 28y = 58x$ ——— (i)

$\Rightarrow 28x + 30y = 58y$ ——— (ii)

solving (i) & (ii), we get: $x = 1, y = 1$

$v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ unit vector of $v = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$

Data compression

unit vector $\times M$

New data points:- $M \times$ unit vector

$$= \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} \times \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 2.12 \\ 2.12 \\ 4.94 \\ 4.94 \end{bmatrix}$$

Steps:-

1. co-variance
2. Eigen value
3. Eigen vector
4. Unit vector
5. Given x unit vector (data compression)

ML (T-20)

24/5/23

Recommender System (Utility Metrix)

→ Ex of an unsupervised learning.

Types of Recommender System :-

① Content Based (user ko interest based)

② Collaborative filtering (user ko jaise similar/connected ones based recommendation)

Collaborative filtering :-

	Item-1	1-2	1-3	1-4
U_1	3	2		
U_2		5	1	2
⋮				
U_n				

→ Utility Matrix

→ Measure similarity

→ Jaccard distance : $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$

→ 2 or vector ko measure distance

$= \frac{1}{2+3-1} = \frac{1}{4}$

→ Cosine similarity : $Sim(A,B) \Rightarrow \cos\theta = \frac{A \cdot B}{|A||B|}$

→ Data normalization.

Ex:

Avg difference in rating :-

Item / customer	A	B	C
JOHN	5	3	2
MARK	3	4	?
JUCY	?	2	5

Item (A,B) = [(5-3)+(3-4)] /2 = 0.5

Item (A,C) = [(5-2)] /1 = 3

Item (B,C) = [(3-2)+(2-5)] /2 = -1

Rating of Jucy (A) :- Rating of A interms of B = 2 + 0.5 = 2.5 (rating of B(JUCY), difference (A,B))

Rating of A interms of C = 5+3 = 8.

Average normalization rating = $\frac{(2.5\times2)+(8\times1)}{3} = 4.33$

Rating of MARK (C) :-

Rating of C interms of A = 3+3 = 6

Rating of C interms of B = 4-1 = 3

Average normalization rating = $\frac{(6\times1)+(3\times2)}{3} = 4$

Cosine Similarity :- (similarity measure)

$Sim(u_1,u_3) = \frac{u_1 . u_2}{\|u_1\|\|u\|} = \cos\theta$

Item / customer	1	2	3	4	5	6	7	8
u_1	5	1	4	?	?		2	1
u_2	5		3	2	5		2	1
u_3	1	4	2	5	2		5	4

Similarity of (u_1,u_2) = 0.99

Similarity of (u_1,u_3) = 0.57

$R(u_2,4)$ $R(u_3,4)$

$R(u_1,4) = \frac{sim(u_1,u_2)\times2 + sim(u_1,u_3)\times5}{sim(u_1,u_2)+sim(u_1,u_3)} = 3.09$

u_1	5	4	2	1
u_2	5	3	2	1

→ 0.99

$R(u_1,5) = \frac{sim(u_1,u_2)\times5 + sim(u_1,u_3)\times2}{sim(u_1,u_2)+sim(u_1,u_3)} = 3.96$

u_1	5	1	1	2	1
u_3	1	4	2	5	4

→ 0.57

ML (T-21)

29/5/23

Recomender System:-

(2) Content based:-

→ Item profile (features of that item)
→ User profile

Ex:- Recommend items to customer x similar to privious items rated highly by x / purchased/ searched by x.

Sln:

Item Profile (set of feature vector)

movies : author, title, actor, ...
Image/Vedio : metadata, tags.
People : set of friends.
text : set of important words (TF-IDF)

User Profile

→ User has rated items with profile $i_1, i_2 \dots i_n$.
→ weight average.

	A	B	C	D
M_1:	1	0	1	0
M_2:	0	1	0	0

Example: Boolean utility matrix (for Movies)

→ Only one feature, 'Actor'.
  → Item profile: vector with 0/1 values.

→ Suppose user x has watched 5 movies.
  → 2 movies featuring actor 'A'
  → 3 movies featuring actor 'B'.

User profile = mean of item profiles
  → feature A's weight = $\frac{2}{5} = 0.4$
  → feature B's weight = $\frac{3}{5} = 0.6$

Item profile of user x

	A	B
M_1:	1	0
M_2:	0	1
M_3:	1	0
M_4:	0	1
M_5:	0	1

User profile

	A	B
u:	0.4	0.6

Cosine Similarity:-

$$sim(u, m_1) = \frac{(0.4 \times 1) + (0.6 \times 0)}{\sqrt{0.4^2 + 0.6^2}\ \sqrt{1^2 + 0}}$$

$$= 0.55$$

$$sim(u, m_2) = \frac{(0.4 \times 0) + (0.6 \times 1)}{\sqrt{0.4^2 + 0.6^2}\ \sqrt{0^2 + 1^2}}$$

$$= 0.83$$

$\therefore M_2 > M_1$.

Information Retrival:- (unsupervised learning, NLP, content based (R.S.))

Ex: Google search, Doc similarity.

*Term frequency $(TF_{ij}) = \frac{f_{ij}}{\text{total no. of terms or word in doc } j}$

where, f_{ij} = frequency of word i in Doc j.

Ex:- Hello Hello everyone

$\frac{\text{Hello}}{2}$ $\frac{\text{everone}}{1}$

$\therefore TF = \frac{2}{3}\ \frac{1}{3}$

*Document Frequency (DF_i) = No. of Docs mention word i

*Inverse Document Frequency $(IDF_i) = \log \frac{N}{DF_i}$

where, N = total no. of Docs

*TF-IDF score: $W_{ij} = TF_{ij} \times IDF_i$

Document profile :- set of words with highest scores (within a given threshold) Together with their score.

	w_1	w_2	w_3	w_4
Doc 1 :	w_{11}	w_{21}	w_{31}	w_{41}
Doc 2 :	w_{12}	w_{22}	w_{32}	w_{42}

Ex:

Doc 1 : Daily Star news

Doc 2 : Daily star post

Doc 3 : Prothom Alo news

Search Query : Daily Daily news.

Sln: search profile

TF :	Daily	Star	news	post	Prothom	Alo
Doc 1 :	$\frac{1}{3}=0.333$	$\frac{1}{3}=0.333$	$\frac{1}{3}=0.333$	0	0	0
Doc 2 :	$\frac{1}{3}=0.333$	$\frac{1}{3}=0.333$	0	$\frac{1}{3}=0.333$	0	0
Doc 3 :	0	0	$\frac{1}{3}=0.333$	0	$\frac{1}{3}=0.333$	$\frac{1}{3}=0.333$
DF_i :	2	2	2	1	1	1
IDF_i :	$\log\frac{3}{2}=0.584$	$\log\frac{3}{2}=0.584$	$\log\frac{3}{2}=0.584$	$\log\frac{3}{1}=1.585$	$\log\frac{3}{1}=1.585$	$\log\frac{3}{1}=1.585$

$TF_{i,j}$-IDF_i:	Daily	Star	news	post	Prothom	Alo
Doc 1 :	0.195	0.195	0.195	0	0	0
Doc 2 :	0.195	0.195	0	0.527	0	0
Doc 3 :	0	0	0.195	0	0.527	0.527

Query Profile:	Daily	news
TF :	$\frac{2}{3}=0.667$	$\frac{1}{3}=0.333$
DF :	2	2
IDF :	$\log\frac{3}{2}=0.584$	$\log\frac{3}{2}=0.584$
TF-IDF:	0.390	0.195

Normalization = $\frac{TF\text{-}IDF}{DF}$

Daily	news
0.195	0.0975

Query Article:

Daily	news
0.195	0.0975

Cosine similarity:

$$sim(Doc1, Q) = \frac{(0.195 \times 0.195) + 0 + (0.195 \times 0.0975) + 0 + 0 + 0}{\sqrt{0.195^2 \times 3}\ \sqrt{0.195^2 + 0.0975^2}}$$

$$= \frac{0.057}{0.338 \times 0.218} = 0.77$$

$$sim(Doc2, Q) = \frac{(0.195 \times 0.195) + 0 + 0 + 0 + 0 + 0 + 0}{\sqrt{0.195^2 + 0.195^2 + 0.527^2}\ \sqrt{0.195^2 + 0.0975^2}}$$

$$= \frac{0.038}{0.595 \times 0.218} = 0.293$$

$$sim(Doc3, Q) = \frac{(0.195 \times 0.0975)}{\sqrt{0.195^2 + 0.527^2 + 0.527^2}\ \sqrt{0.195^2 + 0.0975^2}}$$

$$= \frac{0.019}{0.770 \times 0.218} = 0.113$$

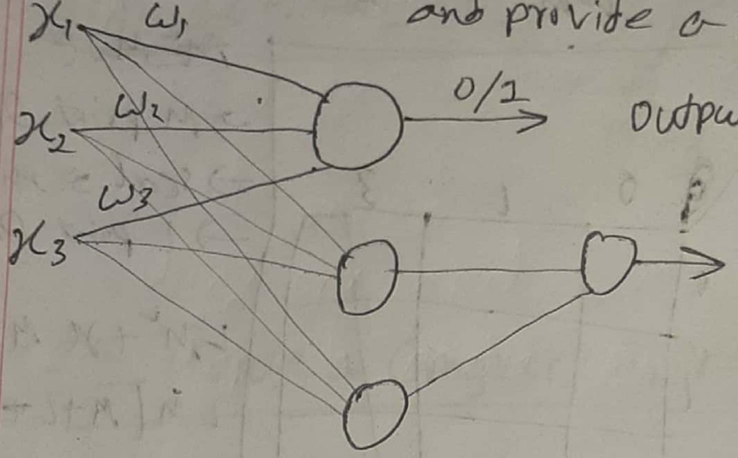
$\therefore D_1 > D_2 > D_3$

ML (T-22)

31/5/23

Neural Network:

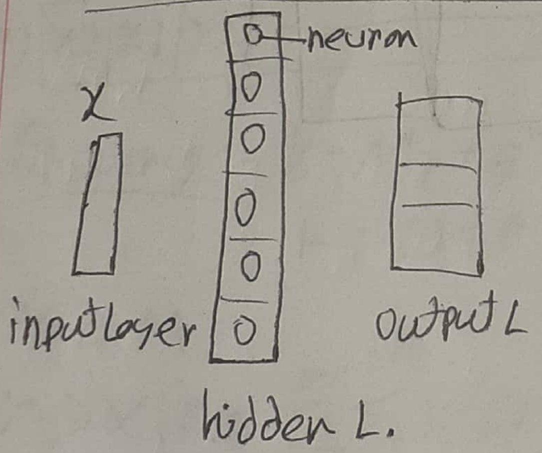
Neuron
↳ Perceptron :- A perceptron takes several inputs $x_1, x_2, \ldots$ and provide a single binary output. → Binary classification



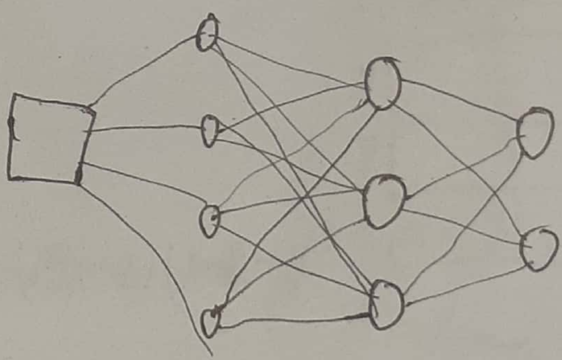
$$\text{output} = \begin{cases} 0, & \text{if } \sum_i \omega_i x_i \leq \text{threshold} \\ 1, & \text{otherwise} \end{cases}$$

threshold হতে কম হলে 0 & বড় হলে 1.
সাধারণত: ω এর মান change হলে: 0/1 এ.

Shallow & Deep Neural Network:



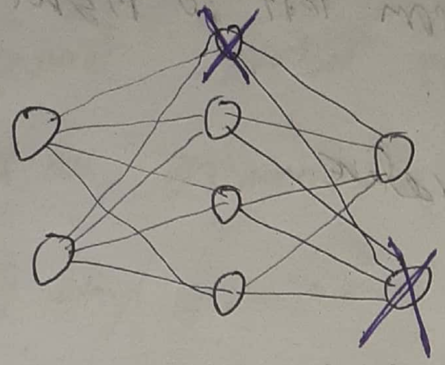
Shallow Network
↳ ১টা hidden layer use



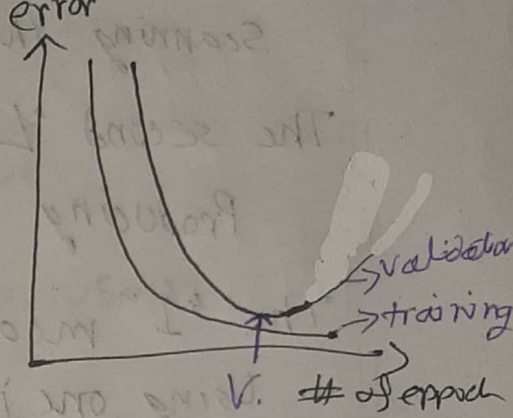
Deep Neural Network

→ Neural Network এর overfit solution

Dropout: প্রতি training এর epoch এ র‍্যান্ডম ভাবে নিউরনগুলি on-off.



→ training data
→ validation data
→ test data



Lab:-

XM + Project Submission
July 10, 2023